



# Shadow volumes

## (Stencil shadows)

En tredje skuggningsmetod.

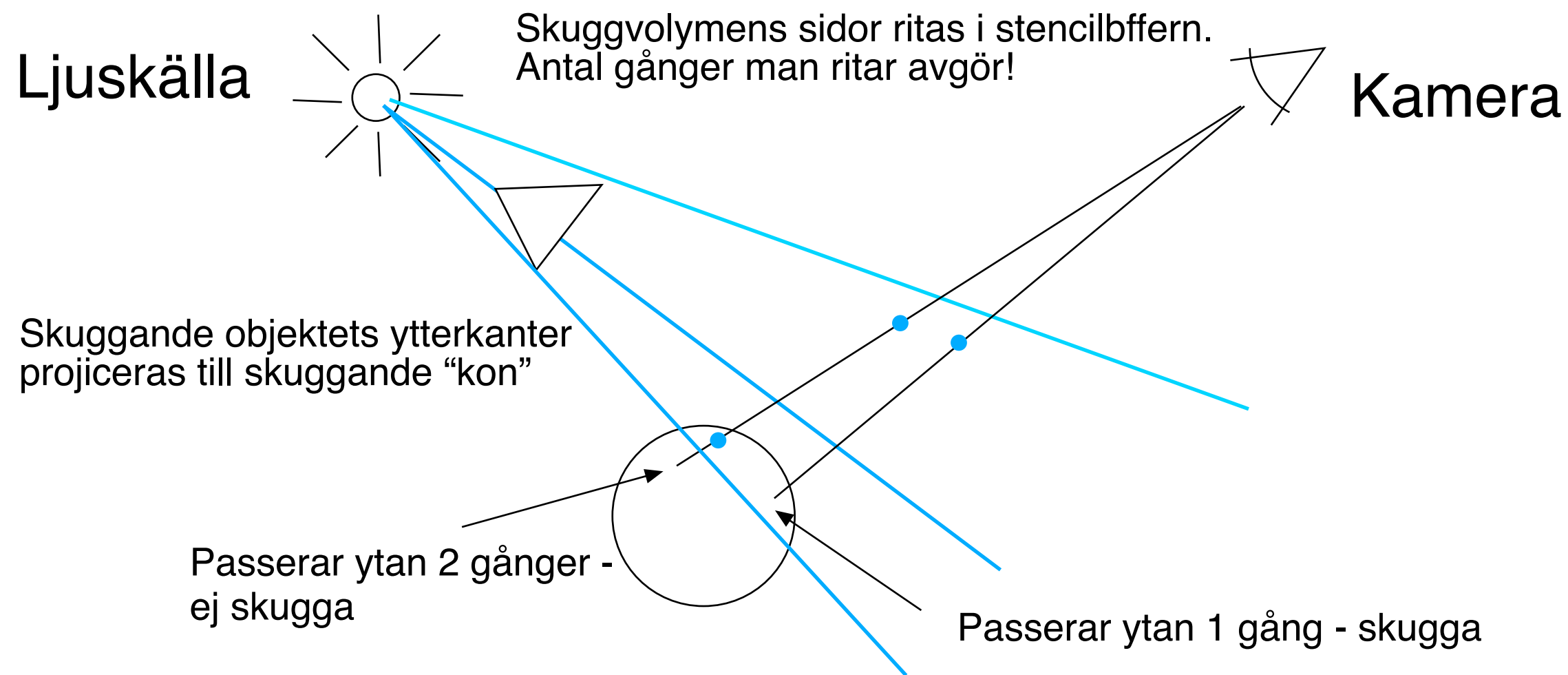
Idé: “projicera” skuggande objekt till kantytter på en volym. Med hjälp av stencilbuffern kan vi avgöra om en viss punkt är inom eller utanför volymen.

Fördel: Undviker upplösningsberoendet som shadow maps har.



# Shadow volumes

## Princip





# Shadow volumes

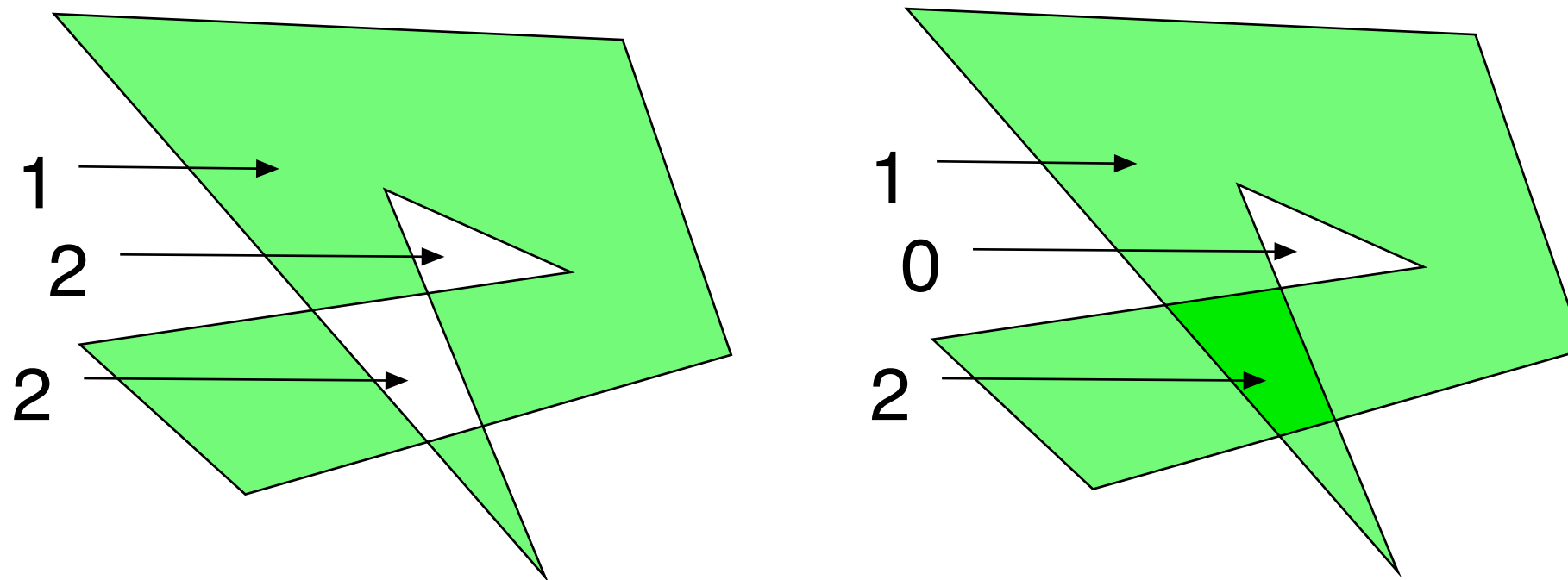
Algoritm:

- Bilda polygoner som utgör sidorna på skuggvolymen
- Initiera stencilbuffern till 1 om kameran är i skuggvolymen, annars 0
  - Rita alla framsidor i stencilbuffern, med inkrement
  - Rita alla baksidor i stencilbuffern, med dekrement
  - Rita scenen
  - Rita skuggor i alla pixlar där stencilbuffern  $> 0$



## Non-zero winding number

Shadow volumes är samma princip i 3D!



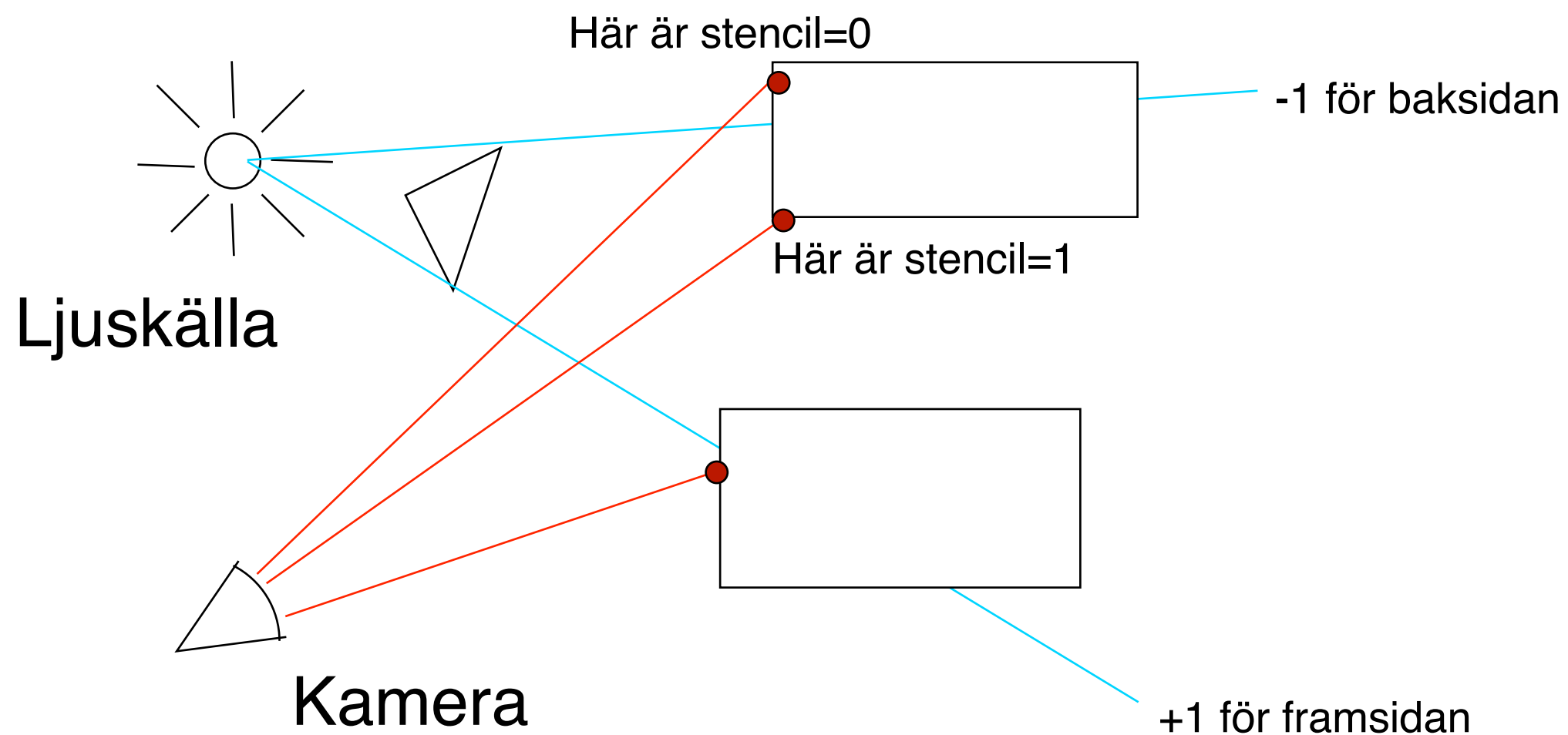
Odd-even rule, udda antal kanter är “innanför” (fel)

Non-zero winding number rule, summan av “upp-kanter” och “ner-kanter” är noll när vi är utanför (rätt)



# Shadow volumes

Stencilbufferns uppdatering och beslut





# Shadow volumes

## Fördelar:

- Inga upplösningsproblem, bra “hårda skuggor”
  - Bygger helt på scencilbuffern
    - Klarar självskuggning
    - Klarar alla riktningar

## Nackdelar:

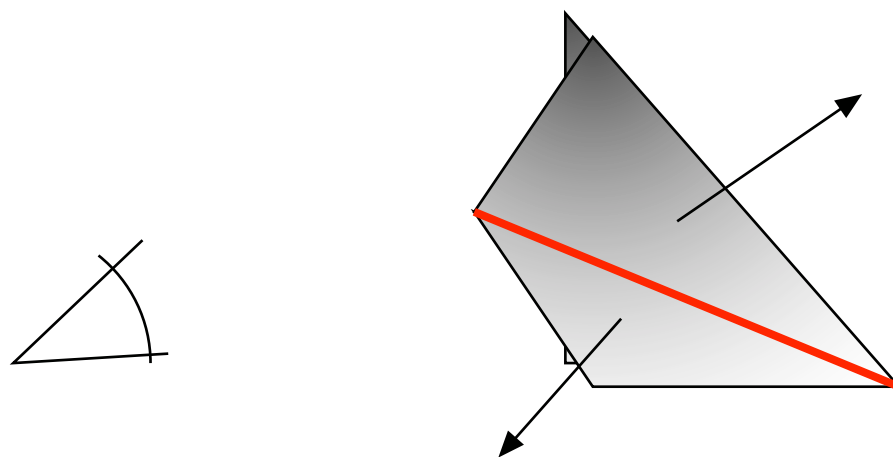
- Bökigt att beräkna volymen
- Många renderingspass över samma yta



# Shadow volumes med hjälp av grannndata

Half-edge eller liknande gör det lättare att hitta skuggvolymen! Sök grannar, detektera kant!

Ett pass över geometrin! Underlättar effektiv generering av skuggvolymen!



Kant = två  
grannpolygoner, en har  
framsida mot kameran,  
en baksida

Men det är ännu bättre än så...



# Shadow volumes helt i GPU

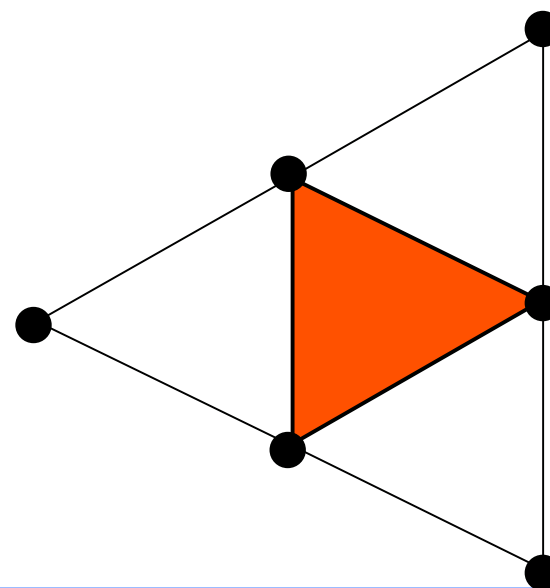
Kan utföras med hjälp av *geometry shaders*.

Granndata kan generera "adjacency"-information för geometry shaders

Adjacency kan med fördel beräknas med hjälp av half-edge.

Genererar skuggvolymen i realtid på GPU!

Nackdelen "bökigt att beräkna volymen" går bort!



Triangel med adjacency:  
Tre grannvertexar följer  
med som del av  
primitivet.





# Shadow volumes

Om vi ser det från skuggkoncepten:

- Receiver: Automatisk! Vi behöver inte identifiera receiver som vi gör med plana skuggor.
- Occluder: Manuell! Occludern måste identifieras och omvandlas till en skuggvolym.
- Mycket bra skarpa skuggor, umbra modelleras exakt. (Vi återkommer till mjuka skuggor.)



# Skuggor

## Sammanfattning

- Projektionsskuggor lätta att göra, men fungerar bara på plana ytor
- Shadow maps fungerar mycket bra, klarar de flesta scener fint, om man bygger en bit vidare från grundmetoden
- Shadow volumes ger fina skuggor, men är scenberoende, kräver specifik processning av skuggande objekt
  - Bortom detta finns mer avancerade metoder för ljusspridning.

Ett tämligen svårt problem som börjar få sin lösning. En självklarhet för högklassig grafik!



# Mer skuggor

Vad mer kan man göra?

Shadow volume BSP trees: En metod för att hantera många skuggande objekt i samma scen

Shadow penumbras: Metoder för att få “mjuka skuggor” i realtid

Det finns mycket att göra. Ändå bygger det mesta på resultat från 70- och 80-talet.



# Flera ljuskällor

Jag har bara tagit upp scener med en ljuskälla.

Grundprincipen för flera ljuskällor är att ljus är additivt! Rendera en gång per ljuskälla, samt ambient separat eller inviktat i en eller flera.

Vissa metoder kan anpassas efter flera ljuskällor i en rendering. (Vilka?)



# Mjuka skuggor

Simulera skuggor från icke punktformiga ljuskällor

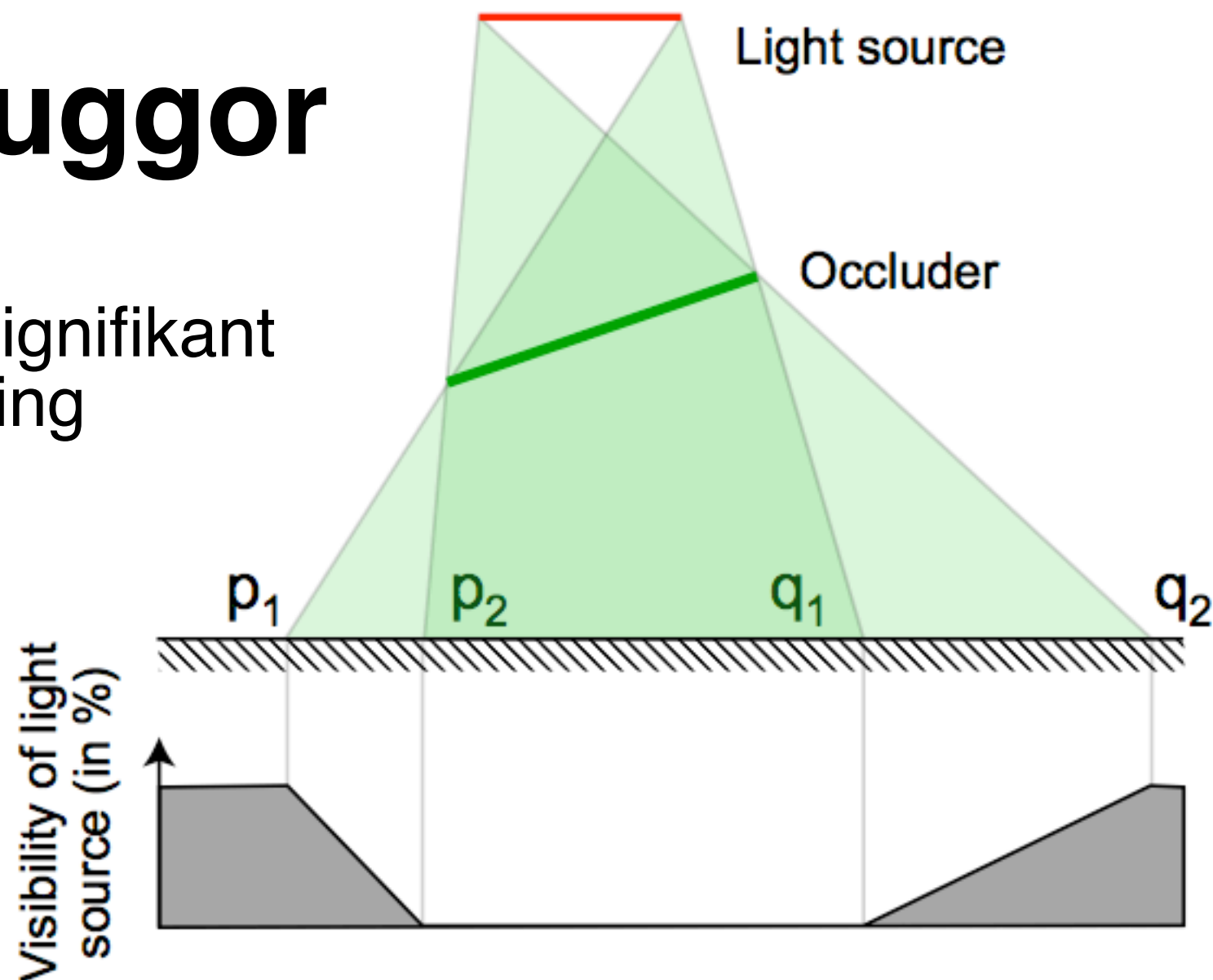
- Multipassrendering
- Percentage Closer Filtering
  - Smoothies
- Soft shadow volumes
- Single sample soft shadows



# Mjuka skuggor

=

ljuskällor med signifikant  
utsträckning



**Figure 12:** *Percentage of a linear light source that is visible.*



# Mjuka skuggor genom multipassrendering

Enkel men ineffektiv lösning

Gör flera renderingar med ljuskällan i olika positioner.

Variant: Rendera enbart skuggor flera gånger.

Kräver många renderingar innan det blir snyggt.  
Omöjlig balans för många scener.



# Percentage Closer Filtering

Vanligaste? Den viktigaste för “allmänbildning” i ämnet.

Relativt enkel metod som beräknar skuggningsgrad ur bilden/skuggmappen.

Uniform filtrering i bildplanet -> penumbrastorleken dåligt avbildad, konstant storlek i bildplanet.





## Percentage Closer Filtering

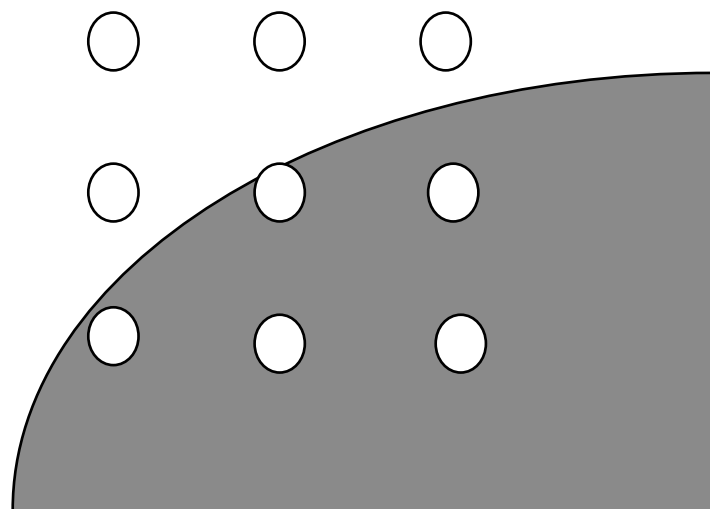
Testa  $n$  punkter kring samplingspunkten.

Om  $k$  av dessa är belysta

belysningsnivå  $k/n$

OBS att bilddata inte filtreras! Räkning av skugga/ej skugga. Materialtextur etc är orört, får enbart ljussättning.

Man kan INTE skapa mjuka skuggor med konventionella lågpasfilter!



4 belysta av 9  $\rightarrow$  ljusnivå  $4/9$

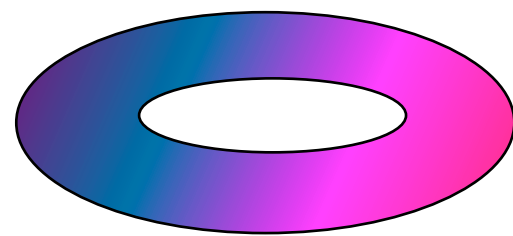


# Smoothies

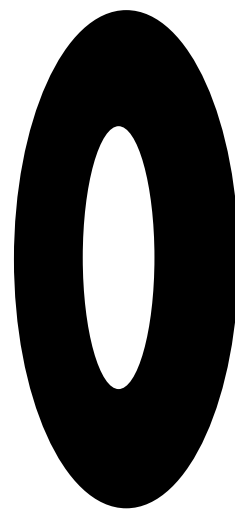
Smoothies “Fake” till och med enligt uppfinnaren!

Yttre penumbra simuleras med extrakanter i utkanten av occluderns projektion.

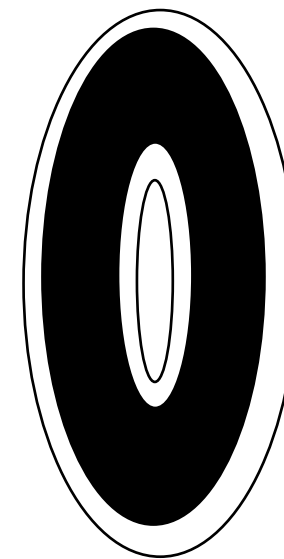
Shadow map + rendering av kanterna.



Objekt



Projicerad profil



“Smoothies” läggs till i kanterna



## **Soft Shadow Volumes**

Soft Shadow Volumes mer ambitiös, multipassmetod med både inre och yttre penumbra. Besläktad med Smoothies.

## **Single Sample Soft Shadows**

Besläktad med PCF. Beräknar avstånd till närmaste skugga. Vissa problem med att närmaste skugga inte alltid är den som har närmast penumbra!

En djungel av algoritmer för mjuka skuggor finns!



# Slutsatser om skuggmetoderna

Mycket viktig aspekt i spel och annan modern realtidsgrafik!

Skuggvolymen ger hög kvalitet men är krävande för komplexa modeller. Scenberoende!

Skuggmappning har i sin enklaste form artefakter och riktningsberoende, men dessa kan hanteras. Ger speciellt bra resultat för mjuka skuggor.

Många algoritmer för mjuka skuggor. Skall man nöja sig med PCF eller gå längre?